

REQUESTS

- XMLHTTP
- FETCH
- AXIOS



AJAX

- ASYNCHRONOUS
- JAVASCRIPT
- AND
- XML

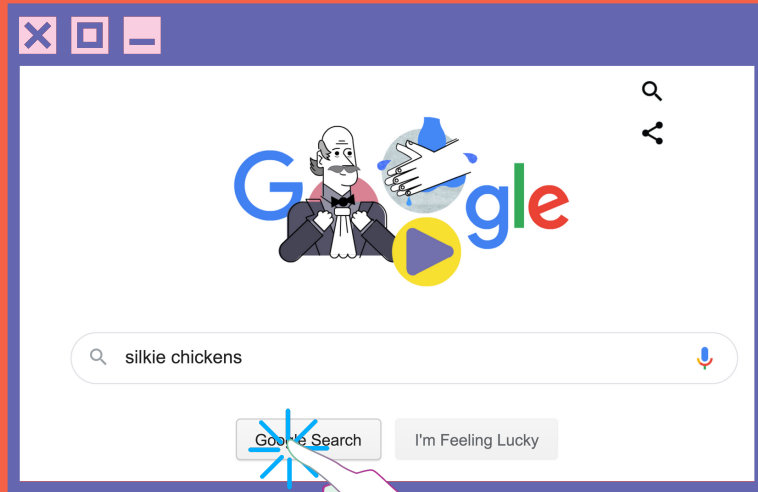


HTTP REQUESTS

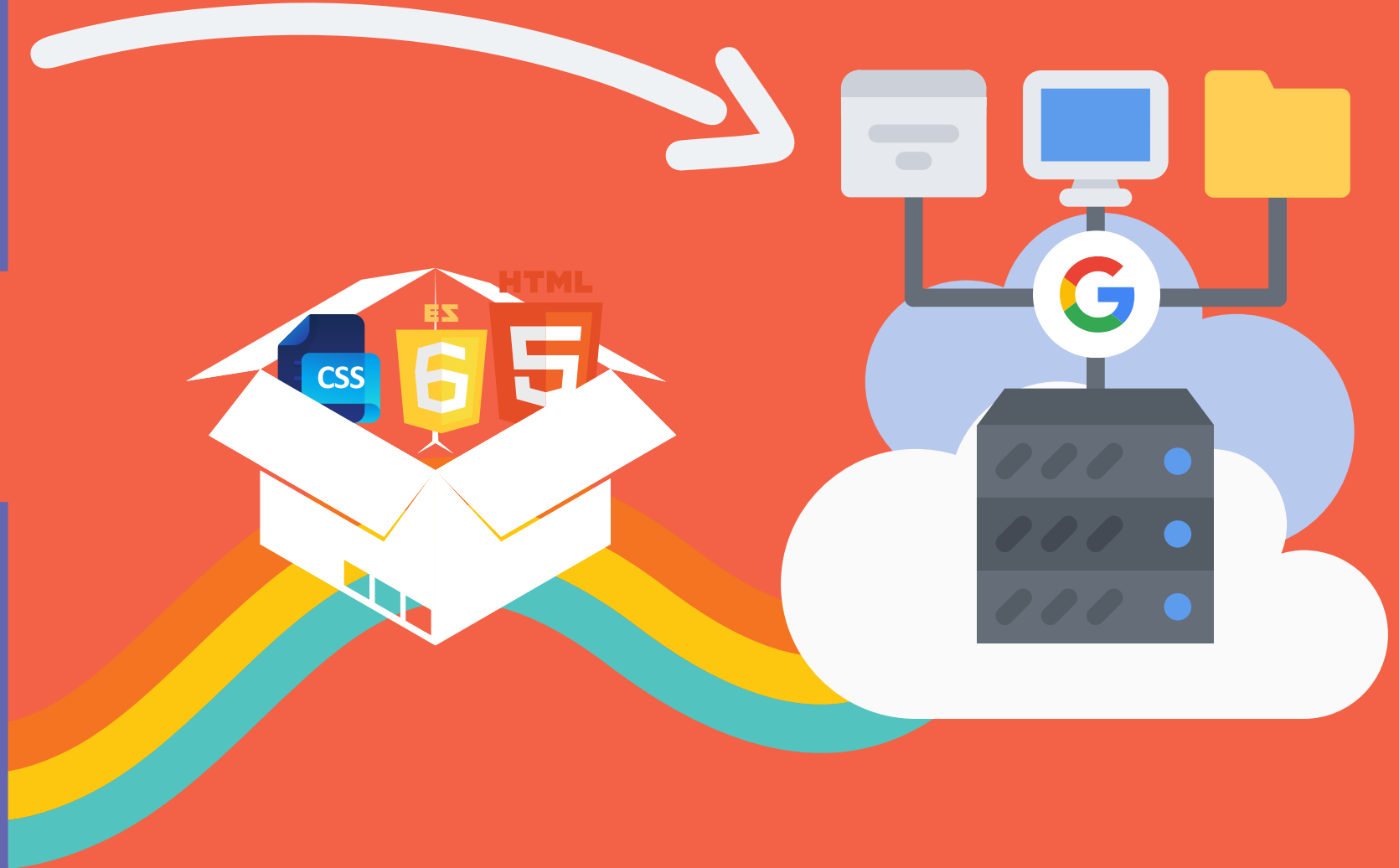
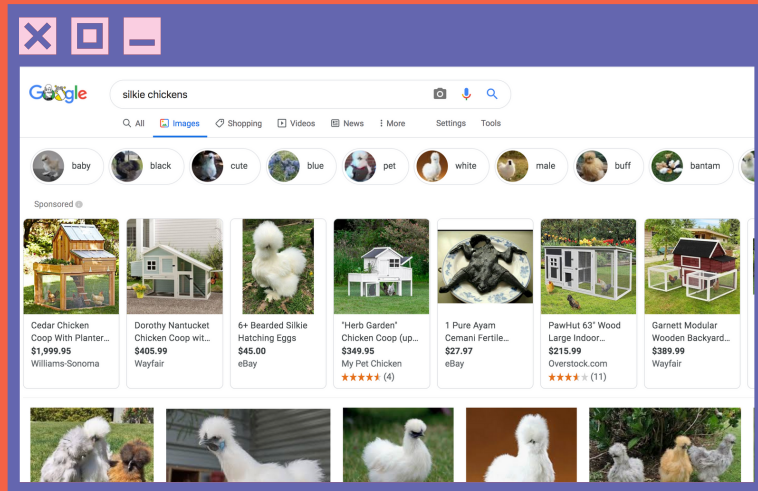


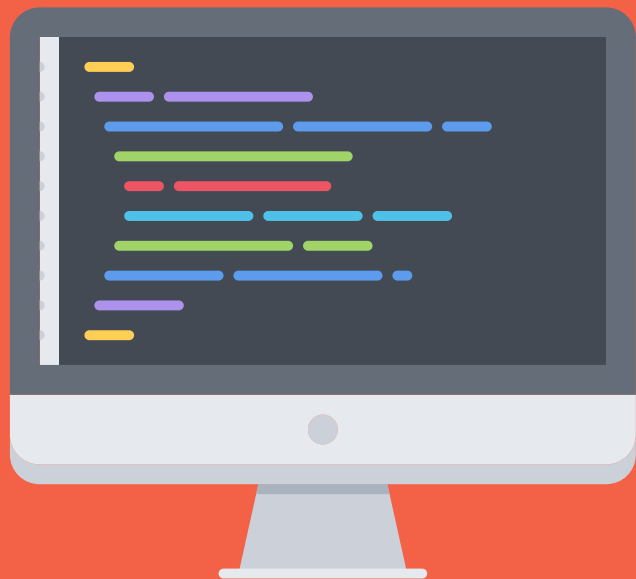
- Foundation of communication on the World Wide Web
- Request -> *I would like this information please*
- "Hyper Text Transfer Protocol"
- Response -> *Ok, here you go!*





PLEASE GIVE ME
GOOGLE.COM/SEARCH?Q=CHICKENS





PLEASE GIVE ME

[API.CRYPTONATOR.COM/API/TICKER/BTC-USD](https://api.cryptonator.com/api/ticker/btc-usd)



price is 11227.76

AJAX

- ASYNCHRONOUS
- JAVASCRIPT
- AND
- XML



AJAJ

- ASYNCHRONOUS
- JAVASCRIPT
- AND
- JSON



JSON

- Java
- Script
- Object
- Notation

```
{  
  "squadName": "Super hero squad",  
  "homeTown": "Metro City",  
  "formed": 2016,  
  "secretBase": "Super tower",  
  "active": true,  
  "members": [  
    "Molecule Man",  
    "Madame Uppercut",  
    "Eternal Flame"  
  ]  
}
```


POST MAN



XMLHttpRequest

- The "original" way of sending requests via JS.
- Does not support promises, so...lots of callbacks!
- WTF is going on with the weird capitalization?
- Clunky syntax that I find difficult to remember!



XMLHttpRequest



```
const myReq = new XMLHttpRequest();

myReq.onload = function() {
  const data = JSON.parse(this.responseText);
  console.log(data);
};
myReq.onerror = function(err) {
  console.log('ERROR!', err);
};
myReq.open('get', 'https://icanhazdadjoke.com/', true);
myReq.setRequestHeader('Accept', ' application/json');
myReq.send();
```

FETCH



Fetch API

- The newer way of making requests via JS
- Supports promises!
- Not supported in Internet Explorer :(

FETCH

```
fetch('https://icanhazdadjoke.com/23/2', {  
  headers : { Accept: 'application/json' }  
})  
  .then((res) => {  
    if (res.status !== 200) {  
      console.log('Problem!', res.status);  
      return;  
    }  
    res.json().then((data) => {  
      console.log(data);  
    });  
  })  
  .catch(function(err) {  
    console.log('Fetch Error', err);  
  });
```

AXIOS

A LIBRARY FOR
MAKING HTTP
REQUESTS

